

图的遍历

广度优先搜索

算法思想

首先访问起点顶点 v ，接着访问 v 的各个邻接顶点 w_1, w_2, \dots ，再从这些访问过的顶点出发，访问它们未被访问过的邻接顶点.....以此类推，直到图中所有的顶点都被访问过为止

特点

广度优先搜索是一个分层查找的过程

每向前走一步可能会访问一批顶点，没有回退情况，不是递归的算法

类似于二叉树的层序遍历，需要借助一个辅助队列

性能分析

需要借助辅助队列，最坏情况下空间复杂度为 $O(V)$

时间复杂度

采用邻接矩阵存储，时间复杂度是 $O(V^2)$

采用邻接表存储，时间复杂度是 $O(V+E)$

应用

原理

广度优先搜索总是按照距离由近到远来遍历图中每个顶点

用BFS求单源最短路径问题

★ 广度优先生成树

邻接矩阵存储表示唯一，广度优先生成树也唯一

邻接表存储不唯一，广度优先生成树也不唯一

深度优先搜索

算法思想

首先访问图中某一起始顶点 v ，由 v 出发，访问与 v 相邻且未被访

问的任一顶点 w_1 ，再访问与 w_1 邻接且未被访问的任一顶点 w_2 ，，，重复上述过程，当不能继续向下访问时，依次回退到最近被访问的顶点，若它还有邻接点未被访问过，则从该点开始继续上述搜索过程，直到所有的顶点都被访问为止

特点

这种搜索算法的搜索策略是尽可能地“深”地搜索一个图

DFS是一个递归算法，类似与树的“先序遍历”，需要借助辅助栈

性能分析

需要辅助栈，空间复杂度 $O(V)$

时间复杂度

采用邻接矩阵存储， $O(V^2)$

采用邻接表存储， $O(V+E)$

深度优先生成树

对连通图调用DFS搜索才能产生深度优先生成树，否则产生的将是深度优先生成森林

基于邻接表存储的深度优先生成树不唯一

BFS与DFS

对于同一个图，采用邻接矩阵存储，遍历得到的DFS,BFS序列是唯一的，采用邻接表存储，遍历得到的DFS,BFS序列是不唯一的

BFS借助队列，DFS借助栈

BFS和DFS的时间复杂度、空间复杂度都相同

BFS一定会生成广度优先生成树、DFS不一定